

AMENDMENTS TO THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) A method for performing a cryptographic function comprising:
calling into an encryption framework to perform the cryptographic function, wherein calling into the encryption framework comprises sending a request to perform the cryptographic function from a kernel consumer; and
processing the request and returning the result to the kernel consumer, wherein processing the request comprises determining whether the request is synchronous or asynchronous, and determining which cryptographic provider to use to perform the cryptographic function, wherein processing the request comprises:
performing the cryptographic function in a kernel consumer context when the request is synchronous;
queuing the request when the request is asynchronous and the kernel consumer indicated to queue the request;
queuing the request when the request is asynchronous and the kernel consumer indicated not to queue the request, and the request needs to be queued;
performing the cryptographic function and returning the result to the kernel consumer after a period of time when the request was queued; and
performing the cryptographic function and returning the result to the request to the kernel consumer when the request is asynchronous, the kernel consumer indicated not to queue the request, and the request does not need to be queued.

2. (Cancelled)

3. (Currently Amended) The method of claim ~~[[2]]~~ 1, wherein queuing the request comprises:
determining whether queue resources are available to queue the request;
notifying the kernel consumer that the request has been successfully queued ~~[[if]]~~ when
queue resources are available; and
notifying the kernel consumer that the request to queue has failed ~~[[if]]~~ when no queue
resources are available and the request is asynchronous,
wherein the kernel customer is one selected from a group consisting of:
notified, and
not notified that the request to the queue has failed when no queue resources are
available and the request is synchronous.
4. (Currently Amended) The method of claim 3, wherein queuing the request further comprises:
~~[[if]]~~ when the request is asynchronous,
registering the kernel consumer to receive a notification when queue resources are
available; and
re-submitting the request by the kernel consumer when the notification is received,
wherein when the request is synchronous, the kernel customer is one selected from
the group consisting of:
registered, and
not registered to receive a notification when the queue resources are
available.
5. (Original) The method of claim 1, wherein the cryptographic provider comprises at least one
selected from the group consisting of a hardware provider and a software provider.
6. (Original) The method of claim 5, wherein the hardware provider is associated with a hardware
provider queue and the software provider is associated with a software provider queue.
7. (Currently Amended) The method of claim ~~[[2]]~~ 1, wherein the request is performed in an
interrupt context ~~[[if]]~~ when the request is asynchronous.

8. (Original) The method of claim 1, wherein the encryption framework is located in a kernel and comprises a kernel interface configured to interface between the encryption framework and the kernel consumer, and a provider interface configured to interface between the cryptographic provider and the kernel interface.
9. (Currently Amended) A method for performing a cryptographic function comprising:
obtaining a request from a kernel consumer by a kernel;
performing the cryptographic function in a kernel consumer context [[if]] when the request is synchronous;
queuing the request [[if]] when the request is asynchronous and the kernel consumer indicated to queue the request;
queuing the request when the request is asynchronous and the kernel consumer indicated not to queue the request, and the request needs to be queued;
performing the cryptographic function and returning the result to the kernel consumer after a period of time [[if]] when the request was queued; and
performing the cryptographic function and returning the result to the request to the kernel consumer [[if]] when the request is asynchronous, the kernel consumer indicated not to queue the request, and the request does not need to be queued.
10. (Original) The method of claim 9, wherein a cryptographic provider performs the cryptographic function.
11. (Original) The method of claim 10, wherein the cryptographic provider comprises at least one selected from the group consisting of a hardware provider and a software provider.
12. (Currently Amended) The method of claim 11, further comprising:
queuing the request [[if]] when the request is synchronous and the hardware provider is required to perform the cryptographic function.
13. (Original) The method of claim 11, wherein the hardware provider is associated with a hardware provider queue and the software provider is associated with a software provider queue.

14. (Currently Amended) The method of claim 9, wherein queuing the request comprises:
determining whether queue resources are available to queue the request;
notifying the kernel consumer that the request has been successfully queued [[if]] when
queue resources are available; and
notifying the kernel consumer that the request to queue has failed [[if]] when no queue
resources are available and the request is asynchronous,
wherein the kernel customer is one selected from a group consisting of:
notified, and
not notified that the request to the queue has failed when no queue resources are
available and the request is synchronous.
15. (Currently Amended) The method of claim 14, wherein queuing the request further comprises:
[[if]] when the request is asynchronous,
registering the kernel consumer to receive a notification when queue resources are
available; and
re-submitting the request by the kernel consumer when the notification is received,
wherein when the request is synchronous, the kernel customer is one selected from
the group consisting of:
registered, and
not registered to receive a notification when the queue resources are
available.
16. (Currently Amended) The method of claim 9, wherein queuing the request further comprises:
returning a handle associated with the request to the kernel consumer [[if]] when queuing the
request is successful and the handle is requested by the kernel consumer.
17. (Original) The method of claim 16, further comprising:
canceling the request using the handle.
18. (Original) The method of claim 9, wherein performing the cryptographic function comprises
generating a context template associated with the request.

19. (Original) The method of claim 18, wherein generating the context template comprises:
 - allocating memory in the kernel for the context template;
 - initializing the context template;
 - computing portions of the context template to obtain computed portions; and
 - storing the context template with the computed portions.
20. (Original) The method of claim 18, wherein the context template is destroyed after all operations specified in the request associated with the context template are completed.
21. (Original) The method of claim 9, wherein the request is a part of a multipart operation.
22. (Original) The method of claim 9, wherein the period of time is related to an amount of time the request remains in a queue associated with the cryptographic provider.
23. (Original) The method of claim 9, wherein the cryptographic function comprises at least one selected from the group consisting of generating a message digest, generating a message authentication code, signature generation and verification, encryption, decryption, and dual-operation routines.
24. (Original) The method of claim 23, wherein dual-operation routines comprise at least one selected from the group consisting of encryption and generating a message authentication code and generating the message authentication code and decryption.
25. (Original) The method of claim 9, wherein the cryptographic function is performed in a plurality of sub-operations.
26. (Original) The method of claim 9, wherein the request comprises a cryptographic mechanism.
27. (Original) The method of claim 26, further comprising:
 - loading the cryptographic provider including the cryptographic mechanism.
28. (Currently Amended) The method of claim 9, wherein the request is performed in an interrupt context [[if]] when the request is asynchronous.

29. (Currently Amended) A computer system for performing a cryptographic function, comprising:

a processor;

a kernel consumer executing on the processor and configured to request the cryptographic function, and

a kernel executing on the processor and comprising:

a cryptographic provider configured to perform the cryptographic function, and

an encryption framework comprising:

a kernel interface configured to interface between the encryption framework and the kernel consumer, and

a provider interface configured to interface between the cryptographic provider and the kernel interface,

wherein the encryption framework is configured to receive and schedule synchronous and asynchronous requests from the kernel consumer;

wherein the encryption framework is configured to process the request and returning the result to the kernel consumer, wherein processing the request comprises:

performing the cryptographic function in a kernel consumer context when the request is synchronous;

queuing the request when the request is asynchronous and the kernel consumer indicated to queue the request;

queuing the request when the request is asynchronous and the kernel consumer indicated not to queue the request, and the request needs to be queued;

performing the cryptographic function and returning the result to the kernel consumer after a period of time when the request was queued; and

performing the cryptographic function and returning the result to the request to the kernel consumer when the request is asynchronous, the kernel consumer indicated not to queue the request, and the request does not need to be queued.

30. (Cancelled)

31. (Cancelled)

32. (Currently Amended) The computer system of claim [[31]] 29, wherein the encryption framework is configured to return a handle to the kernel consumer [[if]] when the request is queued, wherein the encryption framework is not configured to return a handle to the kernel consumer when the request is not queued.

33. (Currently Amended) The computer system of claim 32, wherein the kernel consumer may cancel the request using the handle.

34. (Currently Amended) The computer system of claim 29, wherein the encryption framework is configured to notify the kernel consumer [[if]] when queue resources are not available.

35. (Currently Amended) The computer system of claim 29, wherein the encryption framework is configured to register the kernel consumer such that the kernel consumer is notified when queue resources are available.

36. (Currently Amended) The computer system of claim 29, wherein the cryptographic provider comprises at least one selected from the group consisting of a hardware provider and a software provider.

37. (Currently Amended) The computer system of claim 36, wherein the encryption framework further comprises:

- a hardware provider queue associated with the hardware provider; and
- a software provider queue associated with the software provider.

38. (Currently Amended) The computer system of claim 29, wherein the kernel interface is configured to notify the kernel consumer when the cryptographic provider has completed performing the cryptographic function.

39. (Currently Amended) The computer system of claim 29, wherein the encryption framework is configured to generate a context template, wherein the context template is associated with the request.
40. (Currently Amended) The computer system of claim 39, wherein the encryption framework is configured to initialize and compute portions of the context template.
41. (Currently Amended) The computer system of claim 29, wherein the request comprises a cryptographic mechanism.
42. (Currently Amended) The computer system of claim 29, wherein the encryption framework further comprises a list of cryptographic mechanisms provided by the cryptographic provider.
43. (Currently Amended) The computer system of claim 42, wherein the encryption framework is configured to load the cryptographic provider into the kernel using the list of cryptographic mechanisms and a cryptographic mechanism in the request.
44. (Currently Amended) The computer system of claim 29, wherein the encryption framework comprises functionality to dynamically add a mechanism.
45. (Currently Amended) The computer system of claim 29, wherein the encryption framework comprises functionality to dynamically remove a mechanism.

46. (Currently Amended) A computer system for performing a plurality of cryptographic functions in a kernel, comprising:

a processor;

a kernel consumer executing on the processor and configured to request one of the plurality of cryptographic functions;

a plurality of cryptographic providers executing on the processor and each configured to perform at least one of the plurality of cryptographic functions; and

an encryption framework executing on the processor and comprising:

a kernel interface configured to interface between the encryption framework and the kernel consumer; and

a provider interface configured to interface between the plurality of cryptographic providers and the kernel interface,

wherein the encryption framework is configured to receive and schedule synchronous and asynchronous requests from the kernel consumer;

wherein the encryption framework is configured to process the request and returning the result to the kernel consumer, wherein processing the request comprises:

performing the cryptographic function in a kernel consumer context when the request is synchronous;

queuing the request when the request is asynchronous and the kernel consumer indicated to queue the request;

queuing the request when the request is asynchronous and the kernel consumer indicated not to queue the request, and the request needs to be queued;

performing the cryptographic function and returning the result to the kernel consumer after a period of time when the request was queued; and

performing the cryptographic function and returning the result to the request to the kernel consumer when the request is asynchronous, the kernel consumer indicated not to queue the request, and the request does not need to be queued.

47. (Currently Amended) The computer system of claim 46, wherein the plurality of cryptographic providers comprises at least one selected from the group consisting of a hardware provider and a software provider.
48. (Currently Amended) The computer system of claim 47, wherein the plurality of cryptographic providers comprises a plurality of hardware providers and a plurality of software providers.
49. (Currently Amended) The computer system of claim 48, wherein the encryption framework further comprises a plurality of hardware provider queues, wherein each of the plurality of hardware provider queues is associated with one of the plurality of hardware providers.
50. (Currently Amended) The computer system of claim 48, wherein the encryption framework further comprises a software provider queue associated with the plurality of software providers.
51. (Currently Amended) The computer system of claim 46, wherein the kernel interface is configured to notify the kernel consumer when one of the plurality of cryptographic providers has completed performing the cryptographic function.
52. (Currently Amended) The computer system of claim 46, wherein the encryption framework is configured to initialize and compute portions of [[the]] a context template.
53. (Currently Amended) The computer system of claim 46, wherein the request comprises a cryptographic mechanism.
54. (Currently Amended) The computer system of claim 46, wherein the encryption framework further comprises a list of cryptographic mechanisms provided by each of the plurality of cryptographic providers.
55. (Currently Amended) The computer system of claim 54, wherein the encryption framework is configured to load at least one of the plurality of cryptographic provider into the kernel based on the list of cryptographic mechanisms and a cryptographic mechanism in the request.

56. (Currently Amended) The computer system of claim 46, wherein the encryption framework comprises functionality to dynamically add a mechanism.
57. (Currently Amended) The computer system of claim 46, wherein the encryption framework further comprises functionality to dynamically remove a mechanism.
58. (Currently Amended) A network system having a plurality of nodes, comprising:
a kernel consumer configured to request a cryptographic function;
a kernel comprising:
a cryptographic provider configured to perform the cryptographic function; and
an encryption framework comprising:
a kernel interface configured to interface between the encryption framework and the kernel consumer; and
a provider interface operatively connected to the kernel interface configured to interface between the cryptographic provider and the kernel interface;
wherein the encryption framework is configured to receive and schedule synchronous and asynchronous requests from the kernel consumer;
wherein the encryption framework is configured to process the request and returning the result to the kernel consumer, wherein processing the request comprises:
performing the cryptographic function in a kernel consumer context when the request is synchronous;
queuing the request when the request is asynchronous and the kernel consumer indicated to queue the request;
queuing the request when the request is asynchronous and the kernel consumer indicated not to queue the request, and the request needs to be queued;
performing the cryptographic function and returning the result to the kernel consumer after a period of time when the request was queued; and

performing the cryptographic function and returning the result to the request to the kernel consumer when the request is asynchronous, the kernel consumer indicated not to queue the request, and the request does not need to be queued,

wherein the kernel consumer executes on any node of the plurality of nodes,
wherein the cryptographic provider executes on any node of the plurality of nodes,
wherein the provider interface executes on any of the plurality of nodes, and
wherein the kernel interface executes on any node of the plurality of nodes.